From: UCL Working Papers, Speech, Hearing and Language, 1996

Learning from the experience of building automatic speech recognition systems

Mark Huckvale

PART I - INTRODUCTION

To a language engineer constructing a human-computer interface, 'performance' is the only important measure of success. It can be measured in terms of words communicated or transaction time or ease-of-use; but in all cases it dominates other measures considered imperative in linguistic science: parsimony, elegance, universality, learnability, or psychological reality. Whereas a linguist is interested in the underlying structure of language that explains how meaning is encoded, or why only certain sentence structures occur, the engineer is only interested in successful communication. To him or her, the underlying structure of language is only interesting to the extent that it makes communication more accurate and more reliable. If the regularities and constraints in language can be captured and expressed in a mathematical formalism that can be exploited to make communication effective, then it doesn't matter that such a formalism 'allows' all kinds of unknown phenomena to occur, or is beyond human cognition. On the other hand, the engineer has a weaker sense of 'understanding' than the linguist, and apart from some primitive language acquisition systems [Gorin, 1995] is content to communicate word strings.

The language engineers' concentration on performance arises from their agenda to build working applications - systems for grammar checking, information retrieval, machine translation, or speech recognition. A system that *works* is more use than a system built on the latest variant of Government and Binding theory that doesn't. And this is the heart of the issue addressed in this paper: *after 40 years of modern linguistics the most successful computational linguistic applications use little modern linguistic knowledge*. Grammar checkers are based on templates, information retrieval systems treat documents as bags of words, speech recognition systems don't know about phrase structure. The reason? Systems built on underlying structures of linguistic competence have lamentable engineering performance. Examples are: speech recognition systems with 15% word accuracy, parsing systems that only provide acceptable syntactic structures for 60% of sentences found in newspapers [Magerman, 1995], or translation systems with outputs that require translators to interpret [Wilks, 1992].

The failure of this 'knowledge-based' approach to building applications is a consequence of the linguists' lack of interest in performance in both senses of the word. Knowledge-based systems are built from pre-existing models of language structure in their representations and processes; there is no effort to deal with real linguistic performances nor to maximise engineering performance. The fact that real communication takes place through a noisy channel with limited memory in limited time has had little influence in linguistics - the opinion is that structural relationships between syntax and semantics are logical and independent of time or availability of processing power. Studies of linguistic competence do not address the engineering issues of communication at all; they do not give an answer to why linguistic communication is fast, robust and seemingly effortless.

This denial of modern linguistic knowledge has not however prevented the engineers from building technological systems for language processing with impressive performance. The approach has been to treat linguistic communication as a syntactic pattern recognition problem and to build simple models with good coverage from a 'warts and all' corpus of linguistic performances. A good example is a speech recognition system for fluent read speech from any speaker, using a vocabulary

of 64,000 words which delivers a word accuracy of better than 90% [Woodland et al, 1995].

This paper investigates the approaches that the language engineers have made to build speech recognition systems. It contrasts the conventional logico-deductive framework for linguistic decoding with the architecture that the engineers (working by trial and error) have found to actually work. On the way lessons will be learned from *what the engineers had to do* to solve particular problems. Addressing the problem of communication independently from the problem of competence has in fact created a set of constraints on the form of competence structures and processes as we shall see.

The rest of the paper is in two parts: in II, I shall give 10 lessons that can be learned from the experience of building speech recognition systems. In part III, I shall propose two small linguistic investigations that could be of practical value for engineers.

PART II - LESSONS

II.1 Real speech is more variable than you think

To recognise a handprinted letter it seems obvious to look for those individual characteristics which differentiate it from other letters: that 'l' is made up from a vertical stroke, that 'o' circumscribes an area of white space. This 'feature extraction' is based on the idea that certain binary properties can be found in the input such that a logical argument will lead to the reliable identification of the letter. Similar approaches were pursued for the phonetic recognition of speech for many years: to identify a segment it seemed necessary to identify a set of characteristics that differentiated it from other segments[1].

This 'obvious' approach has been shown to be deeply flawed for speech recognition; and even in handwriting recognition, where the underlying assumptions may be more valid, it is an approach that is losing popularity [Taxt and Olafsdottir, 1990]. I will try to give the argument against feature extraction in words, then describe the engineering alternative.

We know that the realisation of a phonological segment (and analogically a handwritten letter) varies according to producer, context, environment and repetition. This first fact means that we shouldn't expect to see all features[2] of a segment all the time - that we will need to combine the features we have, or even model how features vary according to the environment in which they occur. The difficulty of using features arises because of the combination of this lack of certainty about feature existence with the fallibility of the pattern recognition schemes for finding features. Features may be absent for two reasons: because they are missing in that realisation, or because the input system failed to classify them correctly. Similarly features may appear to be present when they shouldn't be: when the input system produces false alarms.

Such a system might still work if the miss-rate and false-alarm rate of the feature detectors is small - say for highly-articulated clean speech. But in real-life, the performance of these detectors is rather poor: even gross features such as voicing only seem to have a hit-rate of 95% (5% misses, 5%

[1] An additional problem was segmentation in the first place, but we leave that until II.6.

[2] Throughout, I am using 'feature' to mean some realised property of the segment, not some phonological feature.

false-alarms) [Ghiselli-Crippa, et al, 1991]. Features characteristic of place are detected with even less reliability [Huckvale, 1995]. When the variability of the features is combined with the weakness of detection, there are no simple methods for weighing up the feature evidence to make a decision. Thus although in principle we might only need a few features of a series of segments (or of syllables) to be able to identify the underlying word uniquely, in practice the fallibility of the features gives rise to multiple word candidates. With only feature presence/absence indications to go on means that we can't even sensibly rank these candidates for likelihood.

The engineering solution to this problem has been in two parts: (i) define a segment by the 'space' which it occupies, and (ii) postpone decisions about identity until higher level constraints can also be satisfied. The first of these is an abandonment of the 'obvious' approach - feature extraction. Rather than suggest there are characteristics of segments which identify them, we suggest instead that there is a multi-dimensional parameter space in which different segments occupy different regions. The axes of this space are simply the measurements we make of the signal and a point in this space is a combination of a set of these measurements over a given time interval. Realisations of segments[3] cluster in this space as points or trajectories, and different segments cluster at different locations. It may even be the normal case that realisations of a single segment occupy more than one region.

The importance of this shift of emphasis from characteristics of a segment to the space in which segments occupy - is that a segment becomes defined in relation to other segments, in terms of the space that it *doesn't* occupy. Furthermore, any observation - that is any point in the space - will be closest to a particular segment region. Indeed we could measure the similarity of an observation to any segment by finding the distances in this space to each segment region (or to a known segment realisation). Notice the paradigm shift: from identifying segments on the basis of their character, to determining the nearest segment in terms of the range of realisations of all segments. There is no 'spirit' to the phonological segment /p/, no invariant characteristics of its realisations that define it independently from other segments. Distinctive realisations of /p/ are simply those that occupy distinct regions in the measurement space. In a language with a different phonology, the occupation of the space differs and the imputed criteria for identifying a segment appears to change.

The second component of the engineering solution is to leave the distances between observations and regions as just a list of probabilities, so that they can be used in the higher level search, rather than make early decisions about the closest or best segment immediately. This makes sense in a recognition system that is expecting errors in the front end. It is not that a system needs a mechanism for recovery from errors, rather it needs to accommodate inadequacies of observation that must always occur. We shall see the importance of this in II.3.

II.2 Fine detail is irrelevant

The shift from platonic features of a segment to the positioning of a segment in a multi-dimensional space has other consequences: importantly, the space can only have a few dimensions. Engineers have found it impossible to build systems which use measurement spaces of hundreds of dimensions, where each dimension represents a different observation of the signal. The reason is again one of weighing up one feature against another, of establishing the most likely origin of a particular set of errorful observations. For a large number of features (these could be simply

[3]Or other phonological units.

parameters of the short-term spectrum) the problem is how to combine evidence: which features (axes) have the most weight? If features 250 and 750 contradict one another how might the conflict be reconciled?

In pattern recognition terms, to work effectively with multiple features is to have a good idea about how features co-vary - how they correlate for each particular segment. Do features 250 and 750 normally occur together for this segment (positively correlated), or perhaps this segment only gives rise to one or the other (negatively correlated). But for 1,000 fine-grained features, we would need 1,000,000 correlations *per segment*. And to estimate these correlations we need large number of repetitions, and worse, we need to repeat these for every speaker, environment and context. There isn't enough speech in the world![4]

The experience of pattern recognition shows that it is very hard to deal with large observation vectors: simply because there are too many possible combinations of fine-grained detail. In high dimensional spaces it is impossible to measure distances because the relative importance of the dimensions cannot be estimated. This is known to mathematicians as 'the curse of dimensionality'. Current speech recognition systems use surprisingly few parameters, about 10 to 16 spectral envelope parameters per 10ms. On the other hand it seems important that these are fairly independent - that the axes of our multidimensional space are orthogonal, that the covariance matrix is diagonal [Hunt et al, 1991].

In speech terms, we can also get a feel for this. We know that even the most fundamental features of a segment can vary from instance to instance (periodicity, intensity, duration), so it must be the case that the fine features will be even more variable. It doesn't seem likely that hidden away in the coarse-grained variability there is fine-grained invariance. Take realisations of schwa as an extreme example. In pairs like "support/sport" or "lightning/lightening" its presence can be reduced to a lengthening effect on a neighbouring consonant. These realisations of schwa have no common properties at all, only the function of syllable nucleus remains. An analogy is that chairs are identifiable as chairs from their gross shape and function, not from the use of dowel joints.

A corollary of this view - that the parametric space of observations only has a few dimensions - is that the key to perceptual invariance of segments is not to be found in auditory processing. Auditory models provide high-dimensionality representations of the speech signal, with all kinds of fine detail, but if the preceding argument is correct it must be the case that only the gross features of that representation are important. This is not to say that auditory modelling is irrelevant for determining the parameter space in which these gross features operate, nor that human performance in speech perception experiments isn't influenced by auditory processing. Just that to characterise the signal for the purposes of identifying the segmental structure of utterances we only need a few basic parameters which can be obtained by far simpler methods than an auditory model.

II.3 Low-level performance is worse than you think

Underlying much discussion about speech perception is the idea that human performance is highly accurate in segmental terms. That listening to speech is like 'hearing' a printed sentence, with every segment isolated and identified correctly.

There are a number of reasons to suspect that in normal situations - outside the laboratory, not

[4]Not enough for humans or machines.

using simple stimuli, nor highly-constrained and artificial identification tasks - human low-level phonetic recognition performance is actually not that good:

- Recognising peoples names, either spoken or handwritten. At times this seems impossible over the telephone.
- Nonsense word identification scores. These are consistently poorer than for real words. [Miller et al, 1951]
- Difference in identification performance between words in high-predictable and low-predictable contexts. [Kalikow et al, 1977]
- Lack of consistency in phonetic transcription even by experts.
- Best pattern recognition performance is only 70% segments correct. [Robinson, 1991]. Surprisingly poor transcription performance for good quality speech.
- 'Phoneme restoration' [Warren et al, 1970]. Listeners fooled into 'hearing' missing segments.

Of course it is very hard to measure human low-level performance - because even if the task was to generate a phonetic transcription of an utterance from an unknown language, there is still information about phonetic variability and syllabic structure that might be borrowed from the listener's native language. (The listener knows about the dynamics and the constraints of articulation in a way that a pattern recognition system doesn't) But to say that a listener uses expectations about articulation to help recognise the utterance, is to simply confirm that bottom-up information is inadequate.

Speech engineers expect poor low-level performance. A phone recognition rate of 70% (without higher-level constraints) is not a disaster because a 70% correct transcription is not generated and then filtered by higher levels. Rather the evidence about possible transcriptions is utilised directly in the recognition of utterance. By postponing phonetic transcription until after the word sequence is identified, poor segment-level performance can be converted to good word-level performance. We shall see how this is done in II.10. But notice that this postponement of decisions means that the transcription of the utterance comes *after* the words have been recognised, and hence looks rather good. This could be an explanation for the apparent human transcription skill.

II.4 Speakers are more similar to others than to themselves

In our model of a perceptual space, occupied by regions of segments where distances are related to the correlations between the parameters that make up the dimensions, how are variabilities in a segment caused by context (coarticulation) and speaker dealt with?

The conventional view is that speaker differences are so large that they need to be 'normalised' to a standard speaker prior to phonetic processing [Disner, 1980]. In speech recognition systems, it is remarkable how little attention is paid to speaker differences: not only are the models of segments built without regard to speaker, the recognition system does not even impose constraints of speaker continuity through an utterance. However, these systems do have and need different models for segments depending on the phonetic context in which they occur. The corollary of this must be that contextual variability is bigger than speaker variability[5].

[5]The logical argument is made more complex by the fact that speech engineers need large amounts of training data, and hence use multiple speakers to obtain it; and that there are simple algorithms for using contextual constraints in recognition but not speaker constraints. However, speech recognition systems do currently work in this way with fair performance and speaker

Better models of speaker variability might be required, but it may be that better models of contextual variation might give a greater payback - in segment models that incorporate constraints of spectral continuity for example [Holmes, 1996]. Non-segmental approaches to recognition are another possibility [Huckvale, 1995]. It is also possible that the choice of the right observation parameters might also reduce speaker variability [Rosner and Pickering, 1995], or that speaker adaptation methods might suffice. By clarifying the relative importance of sources of variability, the engineers are also able to identify where further gains are likely to be made.

II.5 The consequence of obscurity is ambiguity

It seems to be a fact about utterances that some sections are 'clearer' than others, that some sections are 'more important', that some sections seem to be 'emphasised' by the speaker and 'attended to' better by the listener.

There are many observers of these phenomena who therefore conclude that recognisers need to target such sections specifically. That lexical access is driven by information from stressed syllables, and hence these are 'landmarks' or 'islands of reliability' to which more attention should be paid [Stevens, 1995]. In the pivot parser approach, it is proposed that syllable onsets are *all* you need for lexical access [Dogil and Braun, 1988].

Speech engineers have not targeted stressed syllables as deserving of more processing effort. Although some engineers have tried multiple models for vowels - in stressed and unstressed positions, the results are not conclusive [Hieronymous et al, 1992]. What engineers have most certainly not done is to put more weight on the results of recognising a stressed syllable rather than on recognising an unstressed one. This may seem odd if stressed syllables are clearer and carry more weight. An alternative view is that the recognition system already does *exactly the right thing* under these circumstances: in clear regions of speech, the system generates a few good hypotheses, while in unclear regions, the system generates many weak hypotheses[6]. Since the circumstance of a few good hypotheses will constrain lexical access much more than the circumstance of many weak hypotheses, a clear region will therefore carry more weight.

Notice that the recognition system does not need to know in advance that a given piece of speech is clear or not. It does not have 'stressed syllable detectors', nor indeed does it make any early and firm decisions about clarity or importance; instead the system adapts automatically to changes in clarity - *by having the appropriate decoding process.*

II.6 Segment your knowledge, not your representation

That speech consists of a sequence of discrete segments is an overwhelming illusion to most people. "How is the word 'cat' made up?" "'k' and 'a' and 't'." Phonological systems are based on phonetic transcription: a system of describing speech sounds as a sequence of symbols from a finite vocabulary (even metrical phonologies have 'slots' for melodic information).

---

independence. If the result seems hard to believe, then consider that the acoustic models of segments disregard pitch, operate only on spectral envelope features, and have a low-level performance of only 70%.

[6]This was pointed out to me by John Bridle.

That words have a regular structure made up from a small inventory of possible units is probably a necessary consequence of having to remember them. We couldn't possibly remember words as just sounds - who could identify 50,000 different sounds?[7] - so we impose a logical organisation which provides sequential ordering and constraints on a very small number of basic distinctions.

But just because the mental organisation has this segmented structure, this does not mean that we need to recognise an utterance as segments. The speech engineers have implemented it this way: the recognition task is to find an underlying segmented representation that is the most likely source of the observed signal. That is, given a model of speech generation that happens to have a segmented input, what do we have to put in to get this utterance out? It is perhaps this odd merging of the continuous and the discrete that makes Hidden Markov Modelling so mysterious to non-engineers. In a hidden Markov chain a sequence of observations are produced by a system of both temporal and spectral organisation. There are a number of discrete states (these are the parts of the structure that are associated with discrete segments) which each model a 'region' in our measurement space (its centre and its size), and the trajectory through the space in time is crudely modelled by a sequence of steps through the states. Both the observations and the steps are modelled by probabilities. Any given continuous observation can be assessed for the likelihood that it could have been generated by the Markov model (in terms of regions and trajectories). Since a phonetic transcription is just a sequence of segments, and that a sequence of Markov models is just a bigger Markov model, any continuous utterance can be assessed for the likelihood that it was generated by a given discrete transcription. There are different ways in which the total probability is calculated, but the simplest to understand is the method of total likelihood, which totals all the different ways in which a Markov model could have generated the observation. In doing this, all possible alignments of the model to the observation are explored - in some cases state N will be associated with the time 0.5s, in others with the time 0.9s. The total probability is calculated without actually segmenting the signal - instead all segmentations are used. We end up knowing the likelihood of a hypothetical transcription without knowing which part of the signal is which segment. This is 'recognition without segmentation using a segmented model of production'.

Analogies from handwriting show this phenomenon clearly: we know that writing is made from 70 or so discrete symbols[8], yet connected handwriting doesn't show boundaries and neither can we segment the handwriting without knowing what segments have taken part (i.e. after recognition) [Huckvale, 1992]

The lesson is twofold: firstly that segments are simple-minded but quite workable as underlying entities for modelling pronunciation variation, and secondly that segmentation is a characteristic of the underlying level and can only be inferred on the signal after recognition. These too are lessons that arise as a consequence of postponing decisions about the nature of the utterance for as long as possible.

II.7 Mediocrity at everything is OK

It would be wrong to take as a lesson from the discussion so far that speech recognition systems have any single linguistic component that describes effectively the necessary phenomena at any one

[7] Nor remember 50,000 articulations in production.

[8] In English.

level. It is more the case that the linguistic knowledge in such systems is barely sufficient. Contemporary systems only use bigram models of word order constraints, single pronunciations for words, no modelling of speaker variability at any level, and so on.

Yet if we are to explain the fair performance of such systems, it must be because although they may be mediocre, they are, to coin a phrase, *comprehensively mediocre*. In other words they are made up from a number of rather indifferent source of knowledge which together cover all aspects of the recognition problem. This comprehensive quality is essential for a system that is robust, that doesn't fail for 'non-grammatical' sentences, for intrusive noises, for odd word pronunciations, for disfluencies.

Consider a comprehensive model of syntax. We have already suggested that current parsers only assign reasonable structures to 60% of newspaper sentences. So such a parser is useless to filter out possible from impossible sentence hypotheses - even if the word accuracy was extremely high, the system would have very poor overall performance if it only relied on the syntactic judgements of modern parsers. This is why speech recognition systems, even today, use n-gram probabilities instead. For a sentence hypothesis, the likelihood of the sentence can be calculated from the word frequencies and the word transition probabilities. All sentences can be processed in this way, and all hypotheses can be ranked as (relatively) good or bad. While a probabilistic scheme may only give good average estimates of worth, a parser only needs to make one poor judgement to ruin a whole recognition.

How can it be that a system with such a paucity of linguistic knowledge can perform so well? Simply because linguistic encoding is robust due to constraints and redundancy. We should expect to do fairly well with only a limited amount of knowledge, because an encoding that relied upon sophisticated deduction would not be robust.

All this does not deny that language is intensely sophisticated in the way it can be used to encode and communicate meaning. Models of the way in which meaning is related to the structure of utterances in context appear to require a complex interaction between the roles of the words with a mental model of the communicative act [Sperber and Wilson, 1995]. Yet to be robust, we require that large portions of the utterances are highly predictable (constrained and redundant). These two observations need not be in conflict - the predictability can be high even when the concepts are complex - providing that the speaker acknowledges the complexity in his/her production. In other words to spend more time and effort (and words) communicating complex ideas.

II.8 Unity is strength!

In II.7 the comprehensive quality of speech recognition systems is emphasised, but this is only half the story. Not only must these systems cover every eventuality at every level, but also the knowledge at those levels must be combined. A problem for the knowledge-based approach is that conventional linguistic components have different roles. Typically these are expressed in analytic terms - to build a phonological structure, to build a syntactic structure, to determine some logical form. As a consequence of these different roles it is hard to balance constraints at the different levels - under errorful input which constraint is best to break: a phonotactic one or a syntactic one?

Perhaps an analogy from physics might drive this point home: underlying many disparate physical phenomena is the concept of energy. Thus physicists do not need separate models of causality to explain the roundness of bubbles, or the cooling of coffee, or the fall of an apple. In each case the systems change to minimise their energy. As a consequence of this unity, complex interactions

across domains of surface energy, heat and gravity can be explained. Without such unity we could not reconcile the separate causes to find out what could happen in a physical process involving all domains (such as a boiling kettle).

If we apply this metaphor to language decoding, it suggests that to reconcile constraints across disparate theories and models of linguistics requires an underlying common purpose to those theories. Speech recognition systems make this common purpose explicit: the role of the theories, models, constraints, and knowledge is to estimate the probability that a certain structural relationship could occur. The theories are set up to test hypotheses - note that the generation of these hypotheses is performed elsewhere. Thus the role of phonology is to predict the likelihood that a phonetic representation is a possible realisation of a word, the role of syntax is to predict the likelihood that a word sequence is a possible realisation of a certain syntactic structure, the role of semantics is to predict that an utterance is a realisation of a certain interpretation. In each case the theories deliver numerical values of probability.[9]

Compare this view to the role of such components in a knowledge-based system: here the role of phonology is to parse transcription, the role of the lexicon is to find words which match the phonological sequence, the role of syntax to erect phrase structure. In modern ASR the task of generating hypotheses is left to the search engine - the knowledge is only there to evaluate hypotheses. Hence ASR systems don't *analyse* the data, instead the knowledge components evaluate all possible hypotheses.

These roles may be far away from the conventional view of linguistic theory, but they are essential for a working speech recognition system. To perform a recognition of an utterance it is essential to weigh up constraints arising at different linguistic levels. To combine the likelihood of a spectral vector given a segment with the likelihood of a noun-phrase starting with an adjective. This is only possible by having a unifying role for the different levels; and in engineering systems, this role is taken by probability calculations of structural likelihood.

II.9 Integration makes a difference

The integration of levels using probability is not just 'one possible solution' to speech recognition. It makes a real difference to performance as experience in building speech recognition systems has shown.

A certain knowledge-based speech recognition system had a rather good front end: from a speech signal input alone, it was able to generate a phonetic transcription with about 70% segmental accuracy. To accommodate the remaining errors, a segment lattice was forwarded to the lexical access component. Unfortunately, the lattice needed to contain so many alternatives to achieve good word coverage that the lexical access component hypothesised a very large number of words starting at each lattice position. Once these had been filtered by the syntactic and semantic components a remarkable effect had occurred: in the best word sequences, less than 15% of the segments were correct. In other words, the segment lattice had been effectively stripped of the right answers by the higher level components.

[9] In a modern speech recognition system, the theory of phonology is undertaken by a pronunciation dictionary, the theory of syntax by N-grams, and a semantic theory by post-processing of the N-best utterance interpretations.

Contrast this with a modern probabilistic system. In fact such systems, operating on the signal alone, and forced to create a phonetic transcription, also only achieve about 70% segmental accuracy. On the other hand, if these systems are allowed to exploit lexical and syntactic constraints in the segmental recognition, the rate jumps to 97% correct. This is just the opposite of the knowledge-based system, and shows that higher-level knowledge makes a real difference. But it only makes a real difference when that knowledge is effectively integrated throughout the recognition process.

Integration of knowledge sources is not a design option but is essential for good performance; in ASR this integration is built on the comprehensive coverage of events and on the commonality of purpose to linguistic knowledge.

II.10 Recognition is optimisation

Perhaps the reason it took so long for integrated, probabilistic speech recognition systems to replace knowledge-based ones is that the mechanisms by which knowledge is represented and used in such systems seems backwards. Modern probabilistic systems contain knowledge about speech production, not speech perception. They contain the probability that a word might follow another word, not the probability that a given utterance might contain that sequence. They use the probability that a segment might give rise to a spectral vector, not the likelihood that a spectral vector is evidence for a particular segment. Recognition in these systems is not a kind of hierarchical analysis of the utterance, rather sequences of spectral vectors are judged for the evidence they bring to an overall interpretation.

The ability to use a speech production model in this way is built on the ability to find, for any input observations of an utterance, the most likely input to the model that would have produced such an utterance. The most likely input is also, of course, the best interpretation given all the knowledge available.

Finding the best input to a model of production requires a quite different point of view of the recognition problem, and requires quite different computational processes. The engineers' approach is simply to search the list of all possible utterances - that is match the spectral vectors to all the possible transcriptions of all possible word sequences. This looks impossible, but by organising the list into a directed graph (where each path through the graph is a different utterance) the searching becomes feasible with the right strategy.

The graph searching strategy, sometime called a Viterbi strategy, is based on a simple principle of optimisation: it relies on finding the best path from the starting node to each node in the graph in turn. The Viterbi trick is to store on each node the best path so far, so that paths of length 2 can be built from the best paths of length 1, or paths of length N can be built from the best paths of length N-1. A map analogy may be useful. If the best route from Birmingham to London goes through Watford, then if the best route from Liverpool to London goes through Birmingham, then it must also go through Watford (since when you've got to Birmingham the best continuation is via Watford). This observation is sometimes called Bellman's principle of optimality. In ASR this simple principle of optimality that underlies the graph search makes the search for the single best utterance highly efficient. We build interpretations of utterances left to right[10], holding at each time frame the best paths so far with the associated probability that the observations so far could have

[10]Except for Tony Robinson, who also recognises from right to left.

been generated by this interpretation so far. By the time we reach the end of the observations the best path is also the best interpretation.

Thus the final lesson we should take from speech recognition systems is that we should not miss the opportunity to apply *all* the knowledge we have at any one time to the decoding of a single utterance. The identification of the lowest allophonic variant can be influenced by semantic context. The best interpretation of an utterance is the one that fits best with all of what we know and what we expect. Thus recognition is really constraint satisfaction, and constraint satisfaction is just optimisation. Speech recognition engineers have perhaps stumbled on a major discovery: not only that everything can make a difference, but that everything can be taken into account in a recognition framework based on optimisation. The size-complexity of such a task needn't prevent the right algorithm on the right kind of processing device searching the entire space of possibilities.

While graph-search looks suspiciously like computer science, constraint satisfaction and optimisation are methods used to 'solve' problems by physical systems, plants and animals all the time [McClelland et al, 1988]. They could equally apply to cognitive processing of language by people.

PART III - CONCLUSIONS

Here is a summary of the 10 lessons in Part II:

1. Real speech is more variable than you think
2. Fie detail is irrelevant
3. Low-level performance is worse than you think
4. Speakers are more similar to others than themselves
5. The consequence of obscurity is ambiguity
6. Segment your knowledge, not your representation
7. Mediocrity at everything is OK
8. Unity is strength!
9. Integration makes a difference
10. Recognition is optimisation

These lessons present a picture of a working speech recognition system which differs in a number of ways from the hierarchy of linguistic processing as conventionally conceived. Perhaps the most radical departures are (i) delayed decision making, whereby segments are not recognised nor delineated prior to utterances, and (ii) a common framework of probability which allows high-level constraints to be balanced against low-level signal information. Systems constructed as a sequence of processing levels fail because they make early errorful decisions from which they cannot recover, and because they cannot balance evidence coming from the signal with evidence coming from linguistic constraints.

The other weaknesses of the knowledge-based approach as shown in the lessons are that they don't model variability, they don't deal correctly with ambiguity, and that they are incomplete. On the other hand, we make no particular claim about the quality of the linguistic information used in ASR systems. Since these systems are based on a segmental phonology, have only one pronunciation per word, don't deal with assimilation and elision effects, don't use prosodic cues, don't know about phrase structure or meaning - these systems too are in considerable need of improvement.

But it is important to stress that the improvement of the engineering systems will come from linguistic knowledge that fits within the framework, not just any useful regularity or constraint that could be modelled. A good example would be in phonology. In a number of research papers I

have tried to design a non-segmental model of acoustic structure which performs even as well as the segmental model [Huckvale, 1995]. So far I have failed because the model makes early decisions about phonetic content and has been unable to provide good estimates of probability. The kind of phonological model that is required is one that assigns probabilities to phonetic structures (how likely would this pronunciation of a word be) which could be combined with a probabilistic production model which could calculate the likelihood of a spectral vector at some position within the phonetic structure. This is a kind of probabilistic phonology that I do not believe is being worked on anywhere in the world. The current alternative used in ASR is simply a pronunciation dictionary with one phone sequence per word, and with each phone have a range of acoustic models according to context.

Similarly one could envisage a model of computational semantics of a different kind from the prevailing view of assigning logical formalisms to sentences. A probabilistic semantics would take a syntactically analysed utterance and return the probability that the utterance had a coherent meaning (possibly given the context of earlier utterances). Note that such a system need not actually determine the meaning - just attempt to estimate the probability that the utterance made sense. Some very crude measures of coherence would go a long way.

Finally, the engineering view is becoming more sophisticated. No-one can deny that it has impressive performance given how little it knows. In this paper I have tried to show that when you get the engineering right, decoding utterances is straightforward with only a little knowledge - because language is robust and because we generate utterances so that they may be easily understood. By combining engineering expertise with the right kind of linguistic knowledge and allowing performance to be the ultimate arbiter, the competence of recognition systems will continue to increase. Even if, as many believe, machines need to embody the complexities of modern linguistics to understand language the way humans do, a little knowledge can nevertheless go a long way.

ACKNOWLEDGEMENTS

This paper is a version of a talk given at Birkbeck College London in April 1996. I am grateful to Gareth Gaskell for inviting me and for the useful feedback given there. I am also grateful to Roel Smits for criticisms.

REFERENCES

Disner, S., (1980) "Evaluation of vowel normalisation procedures", JASA 67 253-61.

Dogil, G., Braun, G., (1988) "The PIVOT model of speech parsing", Veroffentlichungen der Komission fur Linguistik und Kommunikationsforschung Nr. 19, Osterreichischen Akademie der Wissenschaften, Wien.

Ghiselli-Crippa, T., El-Jaronidi, A., (1991), "A fast neural network training algorithm and its application to voiced-unvoiced-silence classification of speech", IEEE Conference Acoustics, Speech and Signal Processing, ICASSP-91, pp441-444.

Gorin, A. (1995) "On automated language acquisition", JASA 97 pp3441-3461.

Hieronymus, J.L., McKelvie, D., McInnes, F.R., (1992) "Use of acoustic sentence level and lexical stress in HSMM speech recognition", IEEE Conference Acoustics, Speech and Signal Processing,

ICASSP-92, pp225-228.

Holmes, W., (1996) "Modelling variability between and within speech segments for automatic speech recognition", Speech Hearing and Language - Work in Progress, 9, University College London Phonetics and Linguistics.

Huckvale, M.A., (1992), "Illustrating speech: analogies between speaking and writing", Speech Hearing and Language - Work in Progress 6, University College London Phonetics and Linguistics.

Huckvale, M.A., (1995), "Phonetics characterisation and lexical access in non-segmental speech recognition", International Congress of Phonetics Sciences, Stockholm, pp4:280-283.

Hunt, M.J., Richardson, S.M., Bateman, D.C., Piau, A. (1991), "An investigation of PLP and IMELDA acoustic representations and their potential for combination", IEEE Conference Acoustics, Speech and Signal Processing, ICASSP-91, pp881-884.

Kalikow, D.N., Stevens, K.N., Elliot, L.L., (1977) "Development of a test of speech intelligibility in noise using sentence materials with controlled word predictability", JASA 91 pp1337-1351.

Magerman, D. (1995) "Statistical decision-tree models for parsing", 33rd annual meeting of the Association for Computational Linguistics, Boston MA.

McClelland, J.L., Rumelhart, D.E., Hinton, G.E., (1988) "The appeal of parallel distributed processing", in Parallel distributed processing eds D.E. Rumelhart and J.L. McClelland, MIT Press.

Miller, G.A., Heise, G.A., Lichten, W. (1951), "The intelligibility of speech as a function of the context of the test materials", Journal of Experimental Psycholology 41, pp329-335.

Robinson, T., (1991), "Several improvements to a recurrent error propagation network phone recognition system", Technical report TR82, Cambridge University Engineering Department.

Rosner, B.S., Pickering, J.B., (1994), Vowel Perception and Production, Chapter 5, Oxford University Press.

Sperber, D., Wilson, D., (1995) Relevance: Communication and Cognition, Oxford, Blackwell.

Stevens, K.N., (1985) "Evidence for the role of acoustic boundaries in the perception of speech sounds", in Phonetic Linguistics: Essays in honour of Peter Ladefoged, ed. V.Fromkin, Academic Press.

Taxt, T., Olafsdottir, J.B., (1990) "Recognition of handwritten symbols", Pattern Recognition 23 pp1155-1166.

Warren, R.M., (1970), "Perceptual restoration of missing speech sounds", Science 167 pp392-393.

Wilks, Y., (1992) "SYSTRAN: it obviously works, but how much can it be improved?" in Computers in Translation, ed John Newton, Routledge London.

Woodland, P.C., Leggetter, C.J., Odell, J.J., Valtchev, V., Young, S.J., "The 1994 HTK large vocabulary speech recognition system", IEEE Conference Acoustics, Speech and Signal

Processing, ICASSP-95, pp73-76.